Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python

Bhumika Gupta, PhD
Assistant Professor, C.S.E.D
G.B.P.E.C, Pauri, Uttarakhand, India

Monika Negi, Kanika Vishwakarma, Goldi Rawat, Priyanka Badhani
B.Tech, C.S.E.D
G.B.P.E.C Uttarakhand, India

ABSTRACT
Twitter is a platform widely used by people to express their opinions and display sentiments on different occasions. Sentiment analysis is an approach to analyze data and retrieve sentiment that it embodies. Twitter sentiment analysis is an application of sentiment analysis on data from Twitter (tweets), in order to extract sentiments conveyed by the user. In the past decades, the research in this field has consistently grown. The reason behind this is the challenging format of the tweets which makes the processing difficult. The tweet format is very small which generates a whole new dimension of problems like use of slang, abbreviations etc. In this paper, we aim to review some papers regarding research in sentiment analysis on Twitter, describing the methodologies adopted and models applied, along with describing a generalized Python based approach.

Keywords
Sentiment analysis, Machine Learning, Natural Language Processing, Python.

1. INTRODUCTION
Twitter has emerged as a major micro-blogging website, having over 100 million users generating over 500 million tweets every day. With such large audience, Twitter has consistently attracted users to convey their opinions and perspective about any issue, brand, company or any other topic of interest. Due to this reason, Twitter is used as an informative source by many organizations, institutions and companies.

On Twitter, users are allowed to share their opinions in the form of tweets, using only 140 characters. This leads to people compacting their statements by using slang, abbreviations, emoticons, short forms etc. Along with this, people convey their opinions by using sarcasm and polysemy.

Hence it is justified to term the Twitter language as unstructured.

In order to extract sentiment from tweets, sentiment analysis is used. The results from this can be used in many areas like analyzing and monitoring changes of sentiment with an event, sentiments regarding a particular brand or release of a particular product, analyzing public view of government policies etc.

A lot of research has been done on Twitter data in order to classify the tweets and analyze the results. In this paper we aim to review of some researches in this domain and study how to perform sentiment analysis on Twitter data using Python. The scope of this paper is limited to that of the machine learning models and we show the comparison of efficiencies of these models with one another.

2. ABOUT SENTIMENT ANALYSIS
Sentiment analysis is a process of deriving sentiment of a particular statement or sentence. It's a classification technique which derives opinion from the tweets and formulates a sentiment and on the basis of which, sentiment classification is performed.

Sentiments are subjective to the topic of interest. We are required to formulate that what kind of features will decide for the sentiment it embodies.

In the programming model, sentiment we refer to, is class of entities that the person performing sentiment analysis wants to find in the tweets. The dimension of the sentiment class is crucial factor in deciding the efficiency of the model.

For example, we can have two-class tweet sentiment classification (positive and negative) or three class tweet sentiment classification (positive, negative and neutral).

Sentiment analysis approaches can be broadly categorized in two classes – lexicon based and machine learning based. Lexicon based approach is unsupervised as it proposes to perform analysis using lexicons and a scoring method to evaluate opinions. Whereas machine learning approach involves use of feature extraction and training the model using feature set and some dataset.

The basic steps for performing sentiment analysis includes data collection, pre-processing of data, feature extraction, selecting baseline features, sentiment detection and performing classification either using simple computation or else machine learning approaches.

2.1 Twitter Sentiment Analysis
The aim while performing sentiment analysis on tweets is basically to classify the tweets in different sentiment classes accurately. In this field of research, various approaches have evolved, which propose methods to train a model and then test it to check its efficiency.

Performing sentiment analysis is challenging on Twitter data, as we mentioned earlier. Here we define the reasons for this:

- **Limited tweet size:** with just 140 characters in hand, compact statements are generated, which results sparse set of features.
- **Use of slang:** these words are different from English words and it can make an approach outdated because of the evolutionary use of slangs.
- **Twitter features:** it allows the use of hashtags, user reference and URLs. These require different processing than other words.
- **User variety:** the users express their opinions in a variety of ways, some using different language in between, while others using repeated words or symbols to convey an emotion.

All these problems are required to be faced in the pre-processing section.

Apart from these, we face problems in feature extraction with less features in hand and reducing the dimensionality of features.

3. METHODOLOGY

In order to perform sentiment analysis, we are required to collect data from the desired source (here Twitter). This data undergoes various steps of pre-processing which makes it more machine sensible than its previous form.

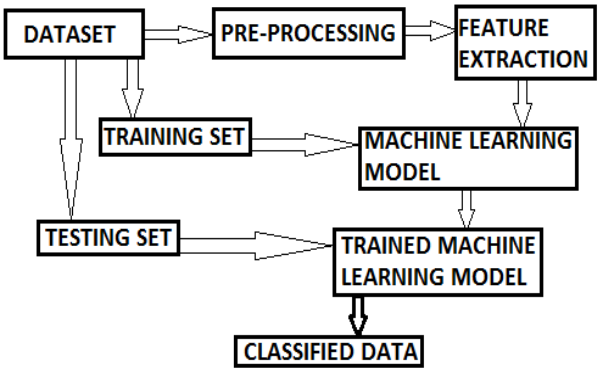

Fig. 1 – General Methodology for sentiment analysis

3.1 Tweet Collection

Tweet collection involves gathering relevant tweets about the particular area of interest. The tweets are collected using Twitter's streaming API [1], [3], or any other mining tool (for example WEKA [2]), for the desired time period of analysis. The format of the retrieved text is converted as per convenience (for example JSON in case of [3], [5]).

The dataset collected is imperative for the efficiency of the model. The division of dataset into training and testing sets is also a deciding factor for the efficiency of the model. The training set is the main aspect upon which the results depends.

3.2 Pre-processing of tweets

The preprocessing of the data is a very important step as it decides the efficiency of the other steps down in line. It involves syntactical correction of the tweets as desired. The steps involved should aim for making the data more machine readable in order to reduce ambiguity in feature extraction. Below are a few steps used for pre-processing of tweets -

- **Removal of re-tweets.**
- **Converting upper case to lower case:** In case we are using case sensitive analysis, we might take two occurrence of same words as different due to their sentence case. It important for an effective analysis not to provide such misgivings to the model.
- **Stop word removal:** Stop words that don't affect the meaning of the tweet are removed (for example and, or, still etc.). [3] uses WEKA machine learning package for this purpose, which checks each word from the text against a dictionary ([3], [5]).
- **Twitter feature removal:** User names and URLs are not important from the perspective of future processing, hence their presence is futile. All usernames and URLs are converted to generic tags [3] or removed [5].
- **Stemming**: Replacing words with their roots, reducing different types of words with similar meanings [3]. This helps in reducing the dimensionality of the feature set.
- **Special character and digit removal:** Digits and special characters don't convey any sentiment. Sometimes they are mixed with words, hence their removal can help in associating two words that were otherwise considered different.
- **Creating a dictionary to remove unwanted words and punctuation marks from the text [5].**
- **Expansion of slangs and abbreviations [5].**
- **Spelling correction [5].**
- **Generating a dictionary for words that are important [7] or for emoticons [2].**
- **Part of speech (POS) tagging:** It assigns tag to each word in text and classifies a word to a specific category like noun, verb, adjective etc. POS taggers are efficient for explicit feature extraction.

3.3 Feature Extraction

A feature is a piece of information that can be used as a characteristic which can assist in solving a problem (like prediction [11]). The quality and quantity of features is very important as they are important for the results generated by the selected model.

Selection of useful words from tweets is feature extraction.

- **Unigram features –** one word is considered at a time and decided whether it is capable of being a feature.
- **N-gram features –** more than one word is considered at a time.
- **External lexicon –** use of list of words with predefined positive or negative sentiment.

Frequency analysis is a method to collect features with highest frequencies used in [1]. Further, they removed some of them due to the presence of words with similar sentiment (for example happy, joy, ecstatic etc.) and created a group of these words. Along with this affinity analysis is performed, which focuses on higher order n-grams in tweet feature representation.

Barnaghi et al [3], use unigrams and bigrams and apply Term Frequency Inverse Document Frequency (TF-IDF) to find the weight of a particular feature in a text and hence filter the features having the maximum weight. The TF-IDF is a very efficient approach and is widely used in text classification and data mining.

Bouazizi et al [4], propose an approach were they don't just rely on vocabulary used but also the expressions and sentence structure used in different conditions. They classified features into four classes: sentiment based features, punctuation and syntax based features, unigram based features and pattern based features.

The work of [5] is a bit different as they don't focus on a particular topic or event but propose to find trending topics in a region. The features extracted are divided in two categories: Common Features and Tweet Specific Features. The former is combination of common sentiment words while the later includes @-network features, user sentiment features and emoticons. Based on the post time of each user, feature vector is built.

3.4 Sentiment classifiers

- **Bayesian logistic regression:** selects features and provides optimization for performing text categorization. It uses a Laplace prior to avoid over-fitting and produces sparse predictive models for text data. The Logistic Regression estimation P(c|f) has the parametric form:

$$\mathbf{P(c|f)} = \frac{\mathbf{1}}{\boldsymbol{z(f)}} \mathbf{exp}\ ((\sum_{i} \boldsymbol{\lambda_{i,c} F_{i,c}(f,c)}))$$

Where $z(f)$a normalization function, λ is is a vector of weight parameters for feature set and $F_{i,c}$ is a binary function that takes as input a feature and a class label. It is triggered when a certain feature exists and the sentiment is hypothesized in a certain way [3].

- **Naïve Bayes:** It is a probabilistic classifier with strong conditional independence assumption that is optimal for classifying classes with highly dependent features. Adherence to the sentiment classes is calculated using the Bayes theorem.

$$P(X|y_i) = \Pi_{i=1}^{m} P(x_i|y_i)$$

X is a feature vector defined as X = $\{x_1, x_2 \ldots . x_m\}$ and y_j is a class label.

Naïve Bayes is a very simple classifier with acceptable results but not as good as other classifiers.

- **Support Vector Machine Algorithm:** Support vector machines are supervised models with associated learning algorithms that analyze data used for classification and regression analysis [6], [9]. It makes use of the concept of decision planes that define decision boundaries.

$$\boldsymbol{g(X)} = \boldsymbol{w^T \emptyset(X) + b}$$

X is feature vector, 'w' is weights of vector and 'b' is bias vector. ∅() is the non-linear mapping from input space to high dimensional feature space. SVMs can be used for pattern recognition [2].

- **Artificial Neural Network:** the ANN model used for supervised learning is the Multi-Layer Perceptron, which is a feed forward model that maps data onto a set of pertinent outputs. Training data given to input layer is processed by hidden intermediate layers and the data goes to the output layers. The number of hidden layers is very important metric for the performance of the model. There are two steps of working of MLP NN- feed forward propagation, involving learning features from feed forward propagation algorithm and back propagation, for cost function [5], [10].

  Zimbra et al [1] propose an approach to use Dynamic Architecture for Artificial Neural Network (DAN2) which is a machine learned model with sufficient sensitivity to mild expression in tweets. They target to analyze brand related sentiments where occurrences of mild sentences are frequent.

  DAN2 is different than the simple neural networks as the number of hidden layers is not fixed before using the model. As the input is given, accumulation of knowledge and learning takes place at each level and forwarded to the next level. The hidden layers are dynamically generated until a desired level of performance is achieved.

- **Case Base Reasoning:** In this technique, problems that were successfully solved in the past are accessed and their solutions are retrieved and used further [10]. It doesn't require an explicit domain model, making elicitation a task of gathering case histories and CBR system can acquire new knowledge as cases. This makes maintenance of large columns of information easier.

- **Maximum Entropy Classifier:** This classifier takes no assumptions regarding the relations between features; it always tries to maximize entropy of a system by computing its conditional distribution of its class labels [9].

$$\boldsymbol{P_\lambda(y|X) = 1/Z(X) exp \sum_{i} \lambda_i f_i(X,y)}$$

'X' is the feature vector and 'y' is the class label. Z(X) is the normalization factor and λ_i is the weight coefficient $f_i(X,y)$ which is the feature function which is defined as

$$f_i(\mathrm{X,x}) = \begin{cases} 1, & \mathrm{X = \ x_i \ and \ y = y_i} \\ 0, & \mathrm{otherwise} \end{cases}$$

- **Ensemble classifier:** This classifier try to make use of features of all the base classifiers to do the best classification. Base classifier used by [9] were Naïve Bayes, SVM and Maximum Entropy. The classifier classifies based on the output of majority of classifiers (voting rule).

4. TWITTER SENTIMENT ANALYSIS WITH PYTHON

4.1 Python

Python is a high level, interpreted programming language, created by Guido van Rossum. The language is very popular for its code readability and compact line of codes. It uses white space inundation to delimit blocks.

Python provides a large standard library which can be used for various applications for example natural language processing, machine learning, data analysis etc.

It is favored for complex projects, because of its simplicity, diverse range of features and its dynamic nature.

4.2 Natural Language Processing (NLTK)

Natural Language toolkit (NLTK) is a library in python, which provides the base for text processing and classification. Operations such as tokenization, tagging, filtering, text manipulation can be performed with the use of NLTK.

The NLTK library also embodies various trainable classifiers (example – Naïve Bayes Classifier).

NLTK library is used for creating a bag-of words model, which is a type of unigram model for text. In this model, the number of occurrences of each word is counted. The data acquired can be used for training classifier models. The sentiment of the entire tweets is computed by assigning subjectivity score to each word using a sentiment lexicon.

4.3 SCIKIT-LEARN

The Scikit-learn project started as scikits.learn, a Google Summer Code project by David Cournapeau. It is a powerful library that provides many machine learning classification algorithms, efficient tools for data mining and data analysis. Below are various functions that can be performed using this library:

- **Classification:** Identifying the category to which a particular object belongs.
- **Regression:** Predicting a continuous-valued attribute associated with an object.
- **Clustering:** Automatic grouping of similar objects into sets.
- **Dimension Reduction:** Reducing the number of random variables under consideration.
- **Model selection:** Comparing, validating and choosing parameters and models.
- **Preprocessing:** Feature extraction and normalization in order to transform input data for use with machine learning algorithm.

In order to work with scikit-learn, we are required to install NumPy on the system.

4.4 NumPy

NumPy is the fundamental package for scientific computing with Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. It contains among other things:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities.

4.5 Setting Up Environment for Sentiment Analysis Using Python

The following components are required to be downloaded and installed properly.

- Download and install Python 2.6 or above in a desired location.
- Download and install NumPy.
- Download and install NLTK library.
- Download and install Scikit-learn library.

4.6 Data Collection

We have two options to collect data for sentiment analysis. First is to use Tweepy - client for Twitter Application Programming Interface (API).

It can be installed using pip command: pip install tweepy

To fetch tweets from the Twitter API one needs to register an App through their Twitter account. After that the following steps are performed:

- Open https://apps.twitter.com/ and click button – 'Create New App'.
- Fill the details asked.
- When the App is created, the page will be automatically loaded.
- Open the 'Keys and Access Tokens' tab.
- Copy 'Consumer Key', 'Consumer Secret', 'Access token' and 'Access Token Secret'.

The keys copied are then inserted into the code, which helps in dynamic collection of tweets every time we run it.

The other option is to gather data non-dynamically using the existing data provided by websites (like kaggle.com) and save the data into whatever format we require (for example JSON, csv etc.).

The former method is slow in nature as it performs tweet collection every time we start the program. The latter approach may not provide us with the quality of tweets we require.

To solve this we can put the code for tweet collection in different module in a way that it doesn't operate every time we run the project.

4.7 Pre-processing in Python

The pre-processing in Python is easy to perform due to functions provided by the standard library. Some of the steps are given below:

- **Converting all upper case letters to lower case.**
- **Removing URLs:** Filtering of URLs can be done with the help of regular expression (http|https|ftp)://[a-zA-Z0-9\\./]+.
- **Removing Handles (User Reference):** Handles can be removed using regular expression - @(\w+) .
- **Removing hashtags:** Hashtags can be removed using regular expression - #(\w+).
- **Removing emoticons:** We can use emoticon dictionary to filter out the emoticons or to save the occurrence of them in a different file.
- **Removing repeated characters.**

4.8 Feature Extraction

Various methodologies for extracting features are available in the present day. Term frequency-Inverse Document frequency is an efficient approach. TF-IDF is a numerical statistic that reflects the value of a word for the whole document (here, tweet).

Scikit-learn provides vectorizers that translate input documents into vectors of features. We can use library function TfidfVectorizer(), using which we can provide parameters for the kind of features we want to keep by mentioning the minimum frequency of acceptable features.

4.9 TRAINING A MODEL

The scikit-library provides various machine learning models whose implementation in code is very easy. For example one can easily create an instance of Support Vector Machine in one line –

```
classifier_poly=svm.SVC()
```

In order to make use of machine learning models, one is required to remember to install NumPy properly and import from scikit-learn the desired model.

After training the model we, use the same instance to test the model and save the results obtained.

5. EXPERIMENTATION FOR MODEL VALIDATION

After the pre-processing and feature extraction steps are performed, we work towards training and validating the model's performance. The collected dataset is divided in two– training set and testing set. The training set is used to train the classifier (machine learned model) while the testing set is the one on which the experimentation is performed. The ratio of training and testing dataset can vary as per to applications. [1] divides the dataset as 70% training and rest testing, whereas [3] which uses cross validation on the dataset by splitting it into 10 sections. This method selects 90% for training set and 10 for testing.

[4] divided the set as training set containing 21000 tweets while testing set 1400 tweets (approx. 93% and 7%) while [5] used 75% data for training set and [9] used approx. 83% for training.

As the classification work in [6] is topic based and adaptive in nature hence excessive manual labeling is avoided which reduces the size of training set.

The model which is chosen for experimentation is trained using the training set of data. Then this same trained model is used to classify new data, by which we can check its accuracy.

As the classification work in [6] is topic based and adaptive in nature hence excessive manual labeling is avoided which reduces the size of training set.

The proposed work of [3] is a bit different as they correlate the event and sentiment by using timestamp. Using this method for a particular event, it's possible to divide it into sub-events and further deepen the study of user sentiments. This approach is complex but yield very detail results when we chose a large event and desire to see fluctuations in user sentiments with time.

The number of classes to be chosen for classification is up to the user. One can perform binary, ternary or multi-class classification based on the type of application we are aiming for. But it has been observed that as the number of classes increases the performance of classifiers decreases [1], [3].

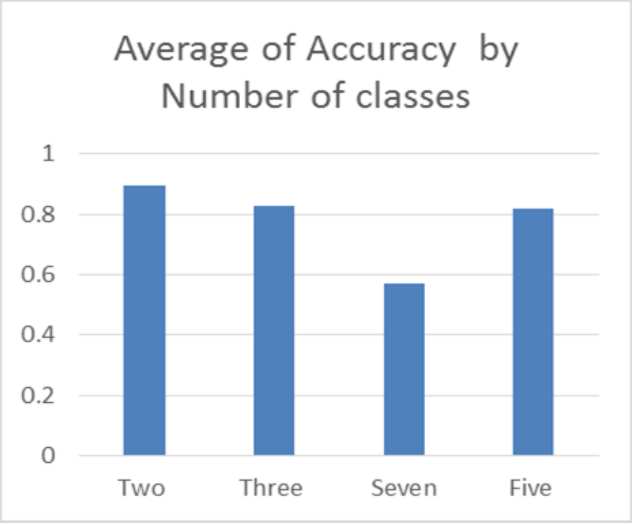

Fig. 2: Accuracy on the basis of number of classes used for classification of sentiments

Table 1: Average accuracies of different models

S. no.	Classifier	Accuracy
1.	DAN2	86.06%
2.	SVM	85.0%
3.	Bayesian Logistic Regression	74.84%
4.	Naïve Bayes	66.24%
5.	Random Forest Classifier	87.5%
6.	Neural Network	89.93%
7.	Maximum Entropy	90.0%
8.	Ensemble classifier	90.0%

5.1 Applications

- **Commerce:** Companies can make use of this research for gathering public opinion related to their brand and products. From the company's perspective the survey of target audience is imperative for making out the ratings of their products. Hence Twitter can serve as a good platform for data collection and analysis to determine customer satisfaction.
- **Politics:** Majority of tweets on Twitter are related to politics. Due to Twitter's widespread use, many politicians are also aiming to connect to people through it. People post their support or disagreement towards government policies, actions, elections, debates etc. Hence analyzing data from it can help is in determining public view.
- **Sports Events:** Sports involve many events, championships, gatherings and some controversies too. Many people are enthusiastic sports followers and follow their favorite players present on Twitter. These people frequently tweet about different sports related events. We can use the data to gather public view of a player's action, team's performance, official decisions etc.

6. CONCLUSION

Twitter sentiment analysis comes under the category of text and opinion mining. It focuses on analyzing the sentiments of the tweets and feeding the data to a machine learning model in order to train it and then check its accuracy, so that we can use this model for future use according to the results. It comprises of steps like data collection, text pre-processing, sentiment detection, sentiment classification, training and testing the model. This research topic has evolved during the last decade with models reaching the efficiency of almost 85%-90%. But it still lacks the dimension of diversity in the data. Along with this it has a lot of application issues with the slang used and the short forms of words. Many analyzers don't perform well when the number of classes are increased. Also it's still not tested that how accurate the model will be for topics other than the one in consideration. Hence sentiment analysis has a very bright scope of development in future.

7. REFERENCES

[1] David Zimbra, M. Ghiassi and Sean Lee, "Brand-Related Twitter Sentiment Analysis using Feature Engineering and the Dynamic Architecture for Artificial Neural Networks", IEEE 1530-1605, 2016.

[2] Varsha Sahayak, Vijaya Shete and Apashabi Pathan, "Sentiment Analysis on Twitter Data", (IJIRAE) ISSN: 2349-2163, January 2015.

[3] Peiman Barnaghi, John G. Breslin and Parsa Ghaffari, "Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment", 2016 IEEE Second International Conference on Big Data Computing Service and Applications.

[4] Mondher Bouazizi and Tomoaki Ohtsuki, "Sentiment Analysis: from Binary to Multi-Class Classification", IEEE ICC 2016 SAC Social Networking, ISBN 978-1-4799-6664-6.

[5] Nehal Mamgain, Ekta Mehta, Ankush Mittal and Gaurav Bhatt, "Sentiment Analysis of Top Colleges in India Using Twitter Data", (IEEE) ISBN -978-1-5090-0082-1, 2016.

[6] Halima Banu S and S Chitrakala, "Trending Topic Analysis Using Novel Sub Topic Detection Model", (IEEE) ISBN- 978-1-4673-9745-2, 2016.

[7] Shi Yuan, Junjie Wu, Lihong Wang and Qing Wang, "A Hybrid Method for Multi-class Sentiment Analysis of Micro-blogs", ISBN- 978-1-5090-2842-9, 2016.

[8] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau, "Sentiment Analysis of Twitter Data" Proceedings of the Workshop on Language in Social Media (LSM 2011), 2011.

[9] Neethu M S and Rajasree R, "Sentiment Analysis in Twitter using Machine Learning Techniques", IEEE – 31661, 4th ICCCNT 2013.

[10] Aliza Sarlan, Chayanit Nadam and Shuib Basri, "Twitter Sentiment Analysis", 2014 International Conference on Information Technology and Multimedia (ICIMU), Putrajaya, Malaysia November 18 – 20, 2014.

[11] Feature engineering, Wikipedia 2017, https://en.wikipedia.org/wiki/Feature_engineering